

SOSYAL TABANLI GÜNCEL SEZGİSEL OPTİMİZASYON ALGORİTMALARI

Soner KIZILOLUK* ve Bilal ALATAŞ**

Özet

Optimizasyon bir işi daha iyi yapma işlemidir. Örneğin optimizasyondaki bir $f(x)$ fonksiyonunda sonuç değerlerini minimum yapacak x değerleri bulmak istenir. Optimizasyon problemlerini çözmek için değişik yöntemler önerilmiştir. Bu yöntemlerden bazılarında doğal süreçlerden esinlenilmiştir. Örnek olarak karınca koloni optimizasyon algoritması verilebilir. Bazı diğer yöntemlerde de sosyal olaylardan esinlenilmiştir. Sosyal tabanlı yöntemlerin sayısı fazla olmamakla birlikte en çok bilineni tabu arama algoritmasıdır. Son zamanlarda ise araştırmacılar öğretim-öğrenme tabanlı algoritma, emperyalist yarışmacı algoritma ve parlamenter optimizasyon algoritmasını geliştirmişlerdir.

Bu çalışmada ise sosyal tabanlı güncel algoritmalar olan emperyalist yarışmacı algoritma ve parlamenter optimizasyon algoritmalarının nasıl modellendiği ve parlamenter optimizasyon algoritmasının problemler üzerinde nasıl uygulandığından bahsedilmiştir.

Anahtar Kelimeler: Emperyalist yarışmacı algoritma, parlamenter optimizasyon algoritması

Current Social-Based Heuristic Optimization Algorithms

Abstract

Optimization is the process of making something better. For example, in an $f(x)$ function in optimization, it is asked for finding x values which make the values of the result minimum. Different methods have been proposed for solving optimization problems. Some of these processes have been inspired by natural processes. Ant colony optimization algorithm can be given as an example. Some of the other methods have been inspired by social events. Although the number of social based methods is limited, the most known one is tabu search algorithm.

* Araş. Gör. Tunceli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü.
sonerkiziloluk@gmail.com

** Yrd. Doç. Dr. Tunceli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü. balatas@tunceli.edu.tr

Recently, researchers have developed teaching-learning based algorithm, imperialist competitive algorithm, and parliamentary optimization algorithm.

In this study, how the imperialist competitive algorithms and parliamentary optimization algorithms which are social-based algorithms are modeled and how parliamentary optimization algorithm is implemented on the problems have been mentioned.

Key Words: Imperialist competitive algorithm, parliamentary optimization algorithm

GİRİŞ

Farklı alanlardaki araştırmacılar doğanın zor problemlerde kullandığı çözümlerden esinlenerek birçok fikir elde etmiştir. Bu alanlardan biri de optimizasyon problemleri için kullanılan çözüm yöntemleridir. Doğanın seçtiği evrimsel yaklaşımların avantajları genetik algoritmalarda ve türevlerinde optimizasyon için önerilmiştir. Hayvanların davranışları; parçacık sürü algoritmaları ve karınca koloni algoritmalarıyla sonuçlanmıştır. Son zamanlarda ise insanların sosyal davranışlarını simüle etme, mühendislikte bazı problemlerin çözümünde yol göstermiştir.

Farklı tipte karar değişkenleri, amaç ve sınırlayıcıların olması durumunda problem formülasyonlarına uygulanabilecek genel çözüm stratejileri sunmaları, kurulması zor olan matematiksel modellere ihtiyaç duymaması, hesaplama güçlerinin iyi ve dönüşümlerinin kolay olması, verilen bir probleme bir çözüm algoritması uyarlamada geçerliliğinin onaylanması zor olabilen bazı varsayımları gerektirmemesi, klasik algoritmadaki gibi ilgilenilen problem üzerinde değişiklik gerektirmemesi vb. avantajlarından dolayı sezgisel algoritmalar yönetim bilimi, bilgisayar, mühendislik gibi birçok farklı alanda yoğun olarak kullanılmakta ve yeni versiyonları önerilmektedir.

Genel amaçlı sezgisel yöntemler; biyolojik tabanlı, fizik tabanlı, sosyal tabanlı, kimya tabanlı ve müzik tabanlı olmak üzere çeşitli gruplara ayrılmaktadır. Ayrıca bunların birleşimi olan melez yöntemler de vardır. Genetik algoritma (GA), diferansiyel gelişim algoritması, karınca koloni algoritmaları, arı koloni algoritmaları ve yapay bağışıklık sistemleri gibi algoritmalar biyolojik tabanlı; ısı işlem ve elektromanyetizma algoritması gibi algoritmalar fizik tabanlı; tabu arama, emperyalist yarışmacı algoritma gibi algoritmalar sosyal tabanlı; yapay kimyasal reaksiyon optimizasyon algoritması kimya tabanlı ve armonik algoritma ise müzik tabanlı algoritmalara örnek olarak verilebilir.(Alataş, 2007).

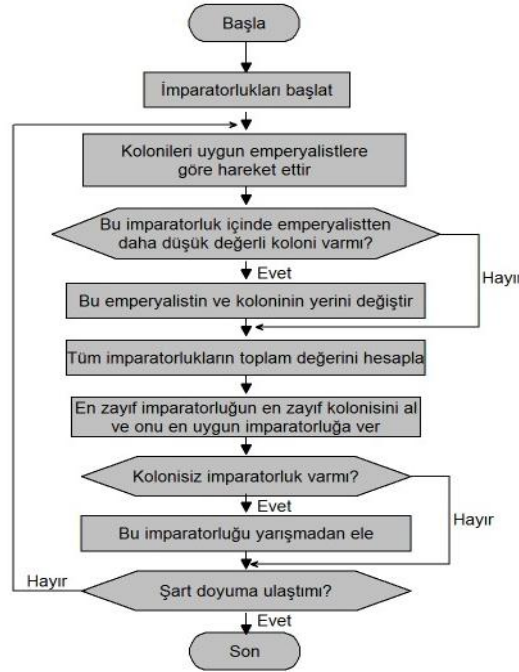
Literatürde dört tane sosyal tabanlı sezgisel optimizasyon algoritması bulunmaktadır. Bunlardan en bilineni ve uygulaması en çok yapılanı tabu arama algoritmasıdır. Yakın zamanda ise üç tane yeni sosyal tabanlı algoritma daha

önerilmiştir: emperyalist yarışmacı algoritma (Atashpaz-Gargari ve Lucas, 2007), parlamenter optimizasyon algoritması (Borji, 2007) ve öğretme-öğrenme tabanlı algoritma (Rao vd., 2012). Henüz yeni önerildiğinden, bu yöntemlerle ilgili çok az sayıda çalışma bulunmaktadır. Hatta parlamenter optimizasyon algoritması ve öğretme-öğrenme tabanlı algoritma ile ilgili sadece ikişer çalışma bulunmaktadır.

Bu makalede temel olarak sosyal tabanlı algoritmalar incelenecektir. Bölüm I'de emperyalist yarışmacı algoritma hakkında genel bilgi verilecektir. Bölüm II'de ise parlamenter optimizasyon algoritması incelenecek ve çalışma mantığı bir örnek uygulama üzerinden gösterilecektir.

I. EMPERYALİST YARIŞMACI ALGORİTMA

Diğer evrimsel algoritmalar gibi Atashpaz-Gargari ve Lucas (2007) tarafından önerilen emperyalist yarışmacı algoritma da bir başlangıç popülasyonu ile başlar (dünyadaki ülkeler). Popülasyondaki birkaç en iyi ülke emperyalist olmak için seçilir ve kalanlar da bu emperyalistlerin kolonisi olur. Tüm koloniler bu emperyalist devletler arasında dağıtılır. Tüm kolonilerin dağıtımından sonra, bu koloniler uygun emperyalistlere doğru hareket etmeye başlar. Bir imparatorluğun toplam gücü emperyalistin ve onun kolonilerinin gücüne bağlıdır. Daha sonra tüm imparatorluklar arasında emperyalistik yarış başlar. Eğer bir imparatorluk bu yarışta gücünü arttıramaz ve başarılı olamazsa yarıştan elenir. Bu yarışta güçlü imparatorluklar gücüne güç katarken zayıf imparatorlukların ise gücü azalır ve sonunda zayıf imparatorluklar yıkılır. Bu yarış en son tek bir imparatorluk kalana kadar devam eder ve sonunda diğer tüm ülkeler bu imparatorluğun bir kolonisi olur. Bu ideal yeni dünyada koloniler emperyalist ile aynı konuma ve güce sahip olacaktır. Şekil 1' de algoritmanın akış şeması görülmektedir (Atashpaz-Gargari ve Lucas, 2007:4662).



Şekil 1: Emperyalist yarışmacı algoritma akış şeması

Emperyalist yarışmacı algoritma güncel bir yöntem olduğu için, bu yöntem ile yapılan çalışma sayısı da oldukça azdır. Analog devre optimizasyonu (Razzaghpour ve Rusu, 2011), birliktelik kurallarının keşfi (Khademolghorani, 2011), gezici robotun global konumlandırılması (Tamimi vd., 2010), kablosuz sensör ağının yerleştirilmesi (Sayadnavard vd., 2010), çevrimiçi PI kontrolü (Movahed ve Yazdani, 2011), DC motorun hız kontrolü (Ghalehpardaz ve Shafiee, 2011), uyarlanabilir emperyalist yarışmacı algoritma (Abdechiri vd., 2010) ve kaotik temelli yapay sinir ağının eğitimi (Abdechiri vd., 2010) emperyalist yarışmacı algoritma ile yapılmış bazı çalışmalardır.

A. BAŞLANGIÇ İMPARATORLUKLARINI ÜRETME

Optimizasyonun amacı problem değişkenlerinden en optimal çözümü bulmaktır. Değer değişkenleri için bir dizi oluşturulur. Genetik algortmada buna “kromozom” adı verilirken emperyalist yarışmacı algortmada “ülke(country)” adı verilir. Bir N_{var} boyutlu optimizasyon probleminde bir ülke bir $1 \times N_{var}$ dizisidir. Bu dizi Denklem 1'deki gibi tanımlanır.

$$country = [u_1, u_2, \dots, u_{Nvar}] \quad (1)$$

Ülkedeki değişken değerleri noktalı kayan sayılarla temsil edilir. Bir ülkenin maliyeti(Cost) $u_1, u_2, \dots, u_{Nvar}$ değişkenlerindeki değer fonksiyonu ile ölçülür.

$$Cost=f(country)=f(u_1,u_2,\dots,u_{Nvar}) \quad (2)$$

Optimizasyon algoritmasına başlamak için N_{pop} boyutundaki başlangıç popülasyonu oluşturulur. İmparatorlukları oluşturmak için N_{imp} sayıdaki en güçlü ülkeler seçilir. Geri kalan N_{col} ise bir imparatorluğa ait olan kolonilerdir. Bu şekilde emperyalist ve koloni olmak üzere 2 tip ülkeye sahip olunur.

Başlangıç imparatorluklarını oluşturmak için koloniler, emperyalistler arasında emperyalistlerin güçlerine göre dağıtılır. Böylece emperyalistlerin başlangıç koloni sayısı gücü ile doğru orantılı olur. Kolonileri emperyalistler arasında doğru orantılı dağıtmak için, bir emperyalistin normalize edilmiş maliyeti Denklem 3'te gösterildiği gibi tanımlanır:

$$C_n = c_n - maks_i\{c_i\} \quad (3)$$

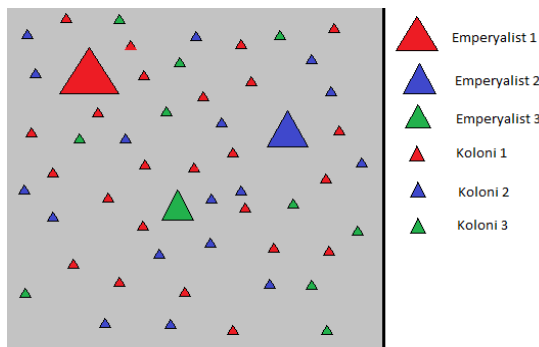
C_n , n . emperyalistin maliyeti ve c_n ise normalize edilmiş maliyettir. Tüm emperyalistlerin normalize edilmiş maliyetlerini içeren, her emperyalistin normalize edilmiş gücü ise Denklem 4'te gösterildiği gibi tanımlanır:

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (4)$$

Bir başka görüşe göre, bir emperyalistin normalize edilmiş gücü, emperyalistin sahip olduğu koloniler kısmıdır. Böylece bir imparatorluğun kolonilerinin başlangıç sayısı Denklem 5'teki gibi olacaktır.

$$N.C._n = round\{p_n.N_{col}\} \quad (5)$$

$N.C._n$ n . İmparatorluğun başlangıç koloni sayısıdır. N_{col} tüm kolonilerin sayısıdır. Kolonileri emperyalistlere dağıtmak için gelişigüzel $N.C._n$ koloni seçilir ve emperyaliste verilir. Bu koloniler emperyalist ile birlikte n . imparatorluğu oluşturur. Şekil 2 her imparatorluğun ilk popülasyonunu gösterir ve daha güçlü imparatorlukların daha çok koloniye sahip olduğu görülebilir (Abdechiri vd., 2010:941).



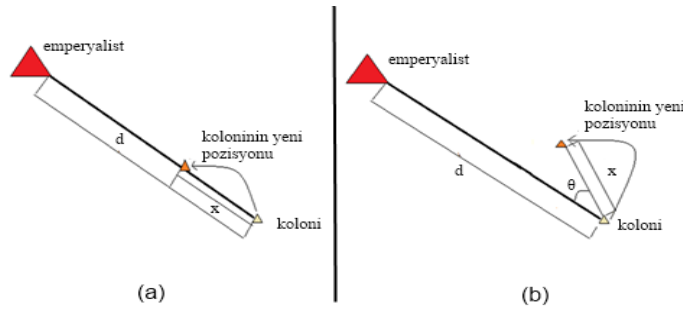
Şekil 2: İmparatorluğun ilk popülasyonu

B. KOLONİLERİN HAREKETİ

Zamanla emperyalist ülkeler kolonilerini arttırmaya başlar. Bu durum kolonilerin emperyaliste doğru hareketi şeklinde gerçekleşir. Şekil 3a'da koloninin x birim emperyaliste hareketi gösterilmektedir. Hareketin yönü koloniden emperyaliste doğru bir vektördür. Şekildeki x gelişigüzel bir değerdir.

$$X \sim U(0, \beta \times d) \quad (6)$$

β 1 den büyük bir sayı ve d ise aradaki uzaklıktır. $\beta > 1$ olması koloninin emperyaliste yaklaşmasına neden olur.



Şekil 3. A) Koloninin emperyaliste hareketi b) Koloninin yeni pozisyonu

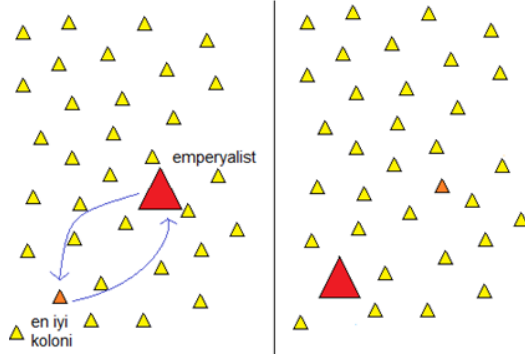
Emperyalistin etrafında farklı noktalar aramak için hareketin yönüne gelişigüzel bir değerde sapma eklenir. Şekil 3b'de θ gelişigüzel bir değerdir.

$$\theta \sim U(-\gamma, \gamma) \quad (7)$$

γ Orijinal yönden sapmanın değerini ayarlayan parametredir. Bununla beraber β ve γ değerleri keyfidir. (Atashpaz-Gargari ve Lucas, 2007:4663).

C. EMPERYALİST VE KOLONİNİN YERİNİ DEĞİŞTİRME

Bir koloni emperyaliste doğru ilerlerken, emperyalistin maliyetinden daha düşük maliyetli bir konuma erişebilir. Böyle bir durumda emperyalist koloni ile yer değiştirir. Daha sonra algoritma, emperyalistin yeni pozisyonu ve kolonilerin bu yeni pozisyona hareketi ile devam eder. Şekil 4, koloni ile emperyalistin yer değiştirmesini gösterir. İmparatorluğun en iyi kolonisi koyu renk ile gösterilmiştir. Bu koloni emperyalistten daha düşük maliyete sahiptir. Şekil 4'te imparatorluğun, emperyalist ile koloninin yer değiştirmesinden sonraki durumu da gösterilmiştir (Atashpaz-Gargari ve Lucas, 2007:4663).



Şekil 4: Koloni ile emperyalistin yer değişmesi

D. İMPARATORLUĞUN TOPLAM GÜCÜ

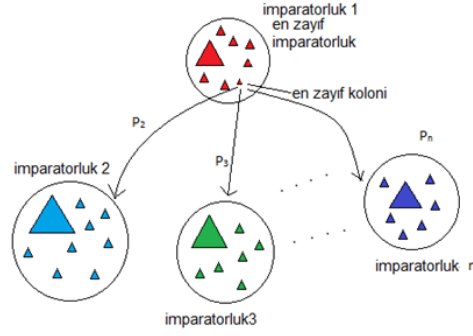
Bir imparatorluğun toplam gücü çoğunlukla emperyalist ülkenin gücünden etkilenir. Fakat imparatorluktaki kolonilerin gücü göz ardı edilebilir olsa da bu imparatorluğun toplam gücü üstünde bir etkiye sahiptir. Bu durum Denklem 8’de gösterildiği gibi toplam maliyeti tanımlayarak modellenmiştir:

$$T.C._n = Cost(imperialist_n) + \xi \text{ mean}\{Cost(colonies of empire_n)\} \quad (8)$$

n . imparatorluğun toplam maliyeti $T.C._n$ ‘dir ve ξ 1’ den küçük olan pozitif bir sayıdır. ξ için küçük bir değer sadece emperyalist tarafından belirlenen imparatorluğun gücünü etkiler ve bu değeri arttırmak imparatorluğun toplam gücüne karar vermede kolonilerin rolünü artırır. 0,1 değeri birçok uygulamada ξ için kullanılmıştır (Atashpaz-Gargari ve Lucas, 2007:4664).

E. EMPERYALİSTİK YARIŞ

Bütün imparatorluklar diğer imparatorlukların kolonilerini ele geçirmeye çalışır ve bunları kontrol eder. Bu emperyalistik yarış gitgide zayıf imparatorlukların gücünde azalmayı ve daha güçlü olanların gücünde artışı beraberinde getirir. Bu yarış, zayıf imparatorlukların zayıf kolonilerinden bazılarının alınması ve diğer imparatorlukların bu kolonileri ele geçirmeye çalışmaları şeklinde modellenmiştir. Şekil 5, modellenmiş emperyalistik yarışın büyük bir resmini gösterir. Bu yarışta imparatorlukların her biri zayıf kolonileri ele geçirme ihtimaline sahiptir. Fakat güçlü imparatorlukların bu kolonileri ele geçirme olasılıkları daha fazla olacaktır.



Şekil 5: Emperyalistik yarışma

Yarışa başlamak için ilk olarak her imparatorluğun toplam gücüne göre ele geçirme olasılığı bulunmalıdır.

$$N.T.C._n = T.C._n - \max_i \{T.C._i\} \quad (9)$$

Denklem 9'daki $T.C._n$ n . imparatorluğun toplam maliyeti ve $N.T.C._n$ normalize edilmiş toplam maliyettir.

$$p_{p_n} = \left| \frac{N.T.C._n}{\sum_{i=1}^{N_{imp}} N.T.C._i} \right| \quad (10)$$

Bahsi geçen kolonileri imparatorlukların sahip olduğu ele geçirme olasılıkları arasında dağıtmak için, Denklem 11'de gösterilen P vektörü oluşturulur.

$$P = [p_{p_1}, p_{p_2}, \dots, p_{p_{N_{imp}}}] \quad (11)$$

Sonra P ile aynı boyutta gelişigüzel sayılardan oluşan R vektörü oluşturulur.

$$R = [r_1, r_2, \dots, r_{N_{imp}}] \\ r_1, r_2, \dots, r_{N_{imp}} \sim U(0, 1) \quad (12)$$

Daha sonra P 'den R 'yi çıkarma işlemi Denklem 13'de gösterilmiştir:

$$D = P - R = [D_1, D_2, \dots, D_{N_{imp}}] = [P_{p_1} - r_1, P_{p_2} - r_2, \dots, P_{p_{N_{imp}}} - r_{N_{imp}}] \quad (13)$$

Vektör D 'ye dayanarak, D 'nin maksimum indeksi ile alakalı olan imparatorluğun kolonileri elde edilir (Atashpaz-Gargari ve Lucas, 2007:4664).

F. BİR NOKTADA BİRLEŞME

Güçsüz imparatorluklar emperyalistik yarışta geride kalır ve çöker. Onun kolonileri ise diğer imparatorluklar arasında paylaşılır. En güçlü imparatorluk

dışındaki bütün imparatorluklar çöktükten sonra tüm koloniler tek bir imparatorluğun kontrolüne girer. Bu ideal yeni dünyada aynı pozisyon ve maliyette olan koloniler, onlarla aynı pozisyon ve maliyete sahip olan bir emperyalist tarafından kontrol edilecektir. Böyle bir durumda emperyalist yarışa son verilmeli ve algoritma durdurulmalıdır (Atashpaz-Gargari ve Lucas, 2007:4664).

II. PARLAMENTER OPTİMİZASYON ALGORİTMASI

İnsan sosyal hayatının birçok durumunda rekabetçi davranışlar gözlemlenebilmektedir. Parlamenter optimizasyon algoritması da genetik algoritmalar ve parçacık sürü algoritmaları gibi olasılıksal, iteratif ve popülasyon tabanlı global optimizasyon tekniğidir. Bilhassa bu metot, parlamentonun kontrolünü ele geçirme çalışmaları sırasındaki grup içi ve gruplar arası çekişmeleri simüle etmeye çalışmıştır.

Parlamentarizm olarak ta bilinen parlamenter sistem, yasaları yapma ve düzenleme gücüne sahip olan hükümet sistemidir. Parlamento üyeleri genel seçimlerde halk tarafından seçilmiştir. İnsanlar genellikle favori partisine oy verirler. Parlamento üyeleri politik partilere üyedirler. Onlar parlamento seçimlerinde partilerini desteklerler. Parlamento üye grupları, ait oldukları partiyi temel alarak, partiler arasındaki yarışmayı diğer partiler üzerinde üstünlük kazanma şeklinde sonuçlandırmak için çalışırlar. Hemen hemen tüm demokratik ülkelerde, politik partiler parlamento popülasyonunu oluştururlar (Borji, 2007; Borji ve Hamidi, 2009).

Parlamento seçimlerinde genelde iki sistem vardır: Çoğunluk seçim sistemi ve orantılı temsil sistemi. Çoğunluk seçim sisteminde her seçim bölgesinden yalnız bir üye seçilir. Orantılı temsil sisteminde bir seçim bölgesinden birkaç üye seçilebilir. Genelde her politik parti aday listelerini sunar ve seçmenler oylayacağı politik parti listesini seçebilir. Partilere aldığı oylarla orantılı olarak parlamentoda sandalye verilir.

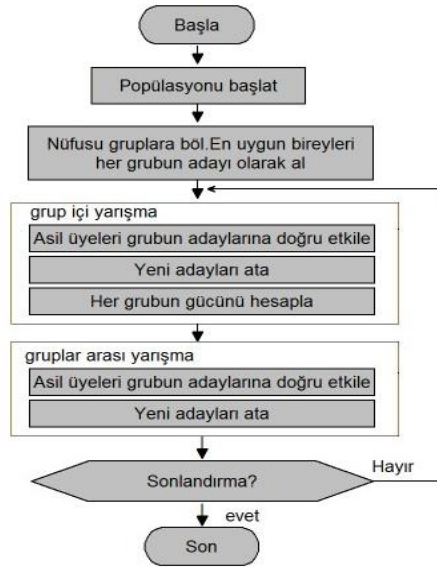
Parlamento içindeki veya dışındaki politik partiler, değişik seviyedeki güçte üyelere sahiptir. Partideki bu insanlar az güç ile diğer asil üyeler üzerinde iyi bir etki bırakmak için uğraşırlar. Bunu onların destekleri ve seçimler sırasındaki oyları için yaparlar. Partinin önemli üyeleri yarışlarda devreye girer ve asil üyeler arasında destek bulmaya çalışır. Diğer bir yandan asil üyeler daha becerikli kişilere eğilimlidir ve genelde inandıkları kişilere oy verirler. Bu süreçte, yüksek kapasiteli genel üyeler önceki adaylarla yerleri değiştirilir. Bu yarışma parti içindeki bireyler arasında olur. Diğer bir yarış ise partiler arasındadır. Partiler daha fazla güç elde etmek için yarışır. Partilerin başarı için iki temel amacı vardır: Parlamentodaki en yüksek sayıdaki sandalyeye sahip olmak ve hükümetin kontrolünü almak (Borji, 2007; Borji ve Hamidi, 2009).

Parlamenter optimizasyon algoritması ise gerçek hayattaki parlamento seçimlerini simüle etmektedir. Algoritmadaki optimizasyon işlemi, ilk olarak birey

popülasyonun oluşturulmasıyla başlar. Bu bireyler parlamentonun üyeleri olarak kabul edilir. Bir sonraki adımda, popülasyon bazı politik gruplar arasında dağıtılır ve yüksek uygunluktaki sabit sayıdaki üyeler grup adayı olarak seçilir.

Popülasyonun bölüştürülmesinden sonra, grup içi yarış başlar. Grup içi yarışmada asil üyeler kendilerine uygun adaylara doğru yönelir. Bu durum asil üye adaylarının vektörlerinin ağırlıklı ortalaması olarak modellenmiştir. Parti içi yarıştan sonra birkaç yüksek uygunluktaki aday grubun son adayları olarak kabul edilir. Bir sonraki adımda bu adaylar diğer grupların adayları ile yarışır. Bir gruptaki adayların ve asil üyelerin her ikisi de grubun toplam gücünün belirlenmesinde önemlidir. Adayların ana gücünün doğrusal kombinasyonu ve asil üyelerin ana gücü, bir grubun toplam uygunluğu olarak nitelendirilir.

Grup içi yarışmadan sonra gruplar arası yarış başlar. Politik gruplar parlamentoda kendi adaylarını kabul ettirmek için diğer gruplarla parlamento yarışı yapar. Bu metotta grubun rolü, bir adayı tanıttıktan sonra hâlâ korunur. Uygunluğu göz ardı edilen grupların gücü aşama aşama azalır ve sonunda çöker. Diğer bir yandan, güçlü gruplar adım adım daha güçlü olmaya başlar ve yarışı kazanmak için daha fazla şans elde eder. Güçlü gruplar bazen birleşmek için anlaşılırlar ve kazanma şanslarını artırırlar. Gruptaki asil üyeler kendi gruplarındaki adaylarla beraber, kendileriyle benzer özellikte ve daha güçlü bir gruba katılırlar. Bu adımlar parlamentoda tek bir grupta birleşilene kadar devam eder. Tüm gruplar birleştikten sonra, asil üyeler lider olan adayla neredeyse eşit güce sahip olurlar. Algoritmanın akış şeması Şekil 6'da gösterilmiştir (Borji, 2007; Borji ve Hamidi, 2009).



Şekil 6. Parlamenter optimizasyon algoritması akış şeması

A. POPÜLASYONUN BAŞLATILMASI

N_{var} boyutundaki başlangıç çözüm popülasyonu, gelişigüzel pozisyonlardaki d boyutlu problem uzayına yayılırlar. Popülasyonun her bireyi boyutsal sürekli vektör olarak kodlanmıştır.

$$P = [p]_1, p_2, \dots, p_n, p_i \in IR \quad (14)$$

Verilen gruptaki bireylerden her biri asil üye ya da aday üye olacaktır. Bir uygun f fonksiyonu bir bireyin gücünü hesaplamak için kullanılır (Borji ve Hamidi, 2009).

B. POPÜLASYONUN BÖLÜMLENDİRİLMESİ

Başlangıç gruplarını oluşturmak için, popülasyon L sayıda bireyden oluşan M tane gruba bölünür.

$$N_{var} = M \times L \quad (15)$$

N_{var}, M ve L pozitif tam sayılar olmak üzere N_{var} Denklem 15'teki gibidir.

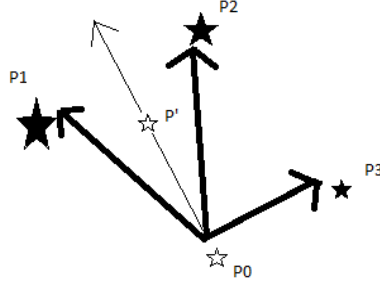
Yüksek uygunluktaki $\theta < L/3$ aday, her grubun adayı olarak nitelendirilir. Bu noktada tüm gruplar eşit sayıda üyeye sahip olurlar, fakat algoritmanın çalışması esnasında gruplar birleşme ve çökme mekanizmasından dolayı farklı sayıda birey elde edebilirler (Borji ve Hamidi, 2009).

C. GRUP İÇİ YARIŞMA

Gruptaki asil üyeler, adaylar ve asil üyeler arasındaki yer alma etkileşiminden sonra, adaylara doğru yönelirler. Bu yönelme, bir üyeyi adaylara bağlayan vektörlerin ağırlıklı ortalamaları ile doğru orantılıdır. Her aday Denklem 16'da gösterildiği gibi kendi aday uygunluklarını arttırmak için ağırlıklandırılmıştır.

$$p' = p_0 + \Pi \left(\frac{(p_1 - p_0) \cdot f(p_1) + (p_2 - p_0) \cdot f(p_2) + (p_3 - p_0) \cdot f(p_3)}{f(p_1) + f(p_2) + f(p_3)} \right) \quad (16)$$

Denklem 16'da p_0 asil üyenin yönelme öncesindeki değeridir. p_1, p_2 ve p_3 aday üyelerin değerleridir. $f(p_1), f(p_2)$ ve $f(p_3)$ ise aday üyelerin uygunluk fonksiyonudur. p' ise asil üyenin aday üyelere doğru yönelmesi sonucundaki aldığı yeni değerdir. Π 0,5 ile 2 arasında gelişigüzel bir sayıdır ve algoritmanın yerel arama alanı çevresindeki adayları aramasına olanak verir. Diğer alternatif bir mekanizmada ise ilk iterasyondan itibaren Π değeri adım adım azaltılır. Şekil 7, yönelme mekanizmasını gösterir (Borji ve Hamidi, 2009).



Şekil 7. Yönlenme mekanizması

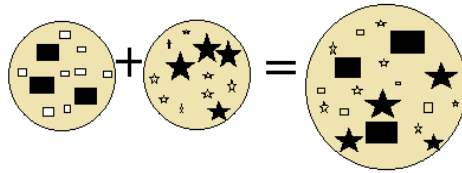
Asil bir üyenin değişmesine izin verilmesi, sadece üye büyük uygunluk değeri aldığıında olur. Yönelmeden sonra asil üyeler, aday üyelerden daha yüksek uygunluk değerine sahip olabilir. Bu gibi durumlarda, adayların yer değişikliği gerçekleşir. $Q^i = [Q_1, Q_2, \dots, Q_\theta]$ adaylar vektörü olsun ve $R^i = [R_{\theta+1}, R_{\theta+2}, \dots, R_i]$ de i . gruptaki geri kalan asil üyeler olsun. Bu grubun gücü Denklem 17'de gösterildiği gibi hesaplanır (Borji ve Hamidi, 2009).

$$power^i = \frac{m \times ort(Q^i) + n \times ort(R^i)}{m + n}; m > n \quad (17)$$

Denklem 17'de $ort(Q^i)$ ve $ort(R^i)$ sırasıyla gruptaki aday üyelerin ve asil üyelerin değerlerinin ortalamasıdır. m ve n ise gelişigüzel değerlerdir. $power^i$ ise i . grubun gücünü ifade eder.

D. GRUPLAR ARASI YARIŞMA

Güçlü gruplar bazen, kendi güçlerini arttırmak için bir gruba katılır ve birleşirler. Birleşmeyi gerçekleştirmek için gelişigüzel bir sayı üretilir ve bu sayı p_m 'den küçük ise, λ sayıda en güçlü grup seçilir ve bir grupta birleştirilir. Şekil 8, bu birleşmeyi göstermektedir. Devam eden algoritma boyunca; zayıf gruplar, hesaplanan gücü korumak ve değer fonksiyonunu azaltmak için silinir. Birleştirmedeki gibi, gelişigüzel bir sayı üretilir ve sayı p_d 'den küçükse, y sayıda minimum güce sahip gruplar elenir (Borji ve Hamidi, 2009).



Şekil 8. Grupların birleşmesi

E. DURUMUN SONLANDIRILMASI

Algoritma sonunda, bir grup yarışı kazanır ve onun en iyi elemanı optimizasyon probleminin çözümü olarak nitelendirilir. İki sonlandırma durumu mevcuttur. Maksimum sayıda iterasyona ulaşıldığında veya bazı başarılı iterasyon sonucunda uygunluk değerinde dikkate değer iyileşme gözlenmezse algoritma sonlandırılır (Borji ve Hamidi, 2009).

F. ÖRNEK BİR PROGRAM

Bu bölümde ise algoritmanın çalışması ile ilgili C# programında örnek bir

uygulama yapılmıştır. Bu uygulamadaki amaç $f(x) = \sum_{i=1}^5 x_i$ fonksiyonunu maksimize etmektir ($0 \leq x_i \leq 100$). Programda gelişigüzel 0 ile 100 arasında değere sahip 5 boyutlu 40 eleman üretilmiştir ve Şekil 9'da görüldüğü gibi başlangıç popülasyonu 5 boyutlu 40 tane elemandır.

0->24 8 39 8 58	20->93 66 97 49 1
1->66 78 92 0 91	21->96 53 1 26 56
2->34 50 7 90 72	22->73 5 69 28 35
3->93 34 71 46 99	23->85 33 86 75 70
4->51 17 95 50 91	24->41 75 62 2 42
5->31 35 72 47 75	25->70 41 68 61 62
6->65 16 39 33 97	26->24 72 45 5 94
7->21 5 4 15 8	27->24 48 22 56 52
8->21 86 84 87 31	28->1 75 9 55 16
9->3 39 90 0 89	29->2 47 59 52 92
10->26 10 39 4 22	30->82 55 84 90 72
11->7 13 88 16 26	31->18 4 94 7 31
12->30 5 44 24 26	32->54 85 40 63 32
13->17 10 73 92 50	33->0 59 64 34 11
14->87 30 56 38 78	34->36 10 29 23 69
15->65 33 7 18 87	35->66 65 6 86 39
16->91 45 72 58 49	36->23 82 15 68 79
17->55 76 35 11 90	37->68 88 55 32 75
18->8 17 88 88 77	38->20 54 15 25 97
19->16 41 60 61 13	39->15 34 45 22 15

Şekil 9: Başlangıç popülasyonunun oluşturulması

Bu elemanlar her grubun eleman sayısı eşit olacak şekilde 4 gruba gelişigüzel olarak dağıtılmıştır. Şekil 10'da elemanların gruplara dağıtıldıktan sonraki hali görülmektedir.

--grup0--		--grup2--	
16	91 45 72 58 49	5	31 35 72 47 75
32	54 85 40 63 32	0	24 8 39 8 58
9	3 39 90 0 89	13	17 10 73 92 50
15	65 33 7 18 87	23	85 33 86 75 70
22	73 5 69 28 35	36	23 82 15 68 79
28	1 75 9 55 16	33	0 59 64 34 11
8	21 86 84 87 31	37	68 88 55 32 75
24	41 75 62 2 42	4	51 17 95 50 91
6	65 16 39 33 97	11	7 13 88 16 26
1	66 78 92 0 91	18	8 17 88 88 77
--grup1--		--grup3--	
21	96 53 1 26 56	12	30 5 44 24 26
38	20 54 15 25 97	10	26 10 39 4 22
2	34 50 7 90 72	31	18 4 94 7 31
27	24 48 22 56 52	25	70 41 68 61 62
3	93 34 71 46 99	35	66 65 6 86 39
30	82 55 84 90 72	14	87 30 56 38 78
29	2 47 59 52 92	17	55 76 35 11 90
26	24 72 45 5 94	19	16 41 60 61 13
34	36 10 29 23 69	7	21 5 4 15 8
20	93 66 97 49 1	39	15 34 45 22 15

Şekil 10: Elemanların gruplara dağıtılması

Daha sonra her elemanın gücünü bulmak için, elemanın boyutlarındaki değerler toplanır. Şekil 11a'da elemanların indeks numaraları ve karşılarında ise uygunluk değerleri (bu uygulama için 5 boyutun değerleri toplamı) görülmektedir. Şekil 11b'de ise her grup elemanı, uygunluk değerine göre kendi içinde büyükten küçüğe sıralanmıştır ve o grubun en büyük değerli üç elemanı ise grubun aday üyeleri olarak belirlenmiştir.

--grup0--		--grup2--		--grup0--		--grup2--	
16->	315	5->	260	28->	156	0->	137
32->	274	0->	137	15->	210	11->	150
9->	221	13->	242	22->	210	33->	168
15->	210	23->	349	9->	221	13->	242
22->	210	36->	267	24->	222	5->	260
28->	156	33->	168	6->	250	36->	267
8->	309	37->	318	32->	274	18->	278
24->	222	4->	304	8->	309	4->	304
6->	250	11->	150	16->	315	37->	318
1->	327	18->	278	1->	327	23->	349
--grup1--		--grup3--		--grup1--		--grup3--	
21->	232	12->	129	34->	167	7->	53
38->	211	10->	101	27->	202	10->	101
2->	253	31->	154	38->	211	12->	129
27->	202	25->	302	21->	232	39->	131
3->	343	35->	262	26->	240	31->	154
30->	383	14->	289	29->	252	19->	191
29->	252	17->	267	2->	253	35->	262
26->	240	19->	191	20->	306	17->	267
34->	167	7->	53	3->	343	14->	289
20->	306	39->	131	30->	383	25->	302

(a)

(b)

Şekil 11: a) Boyutlar toplamı b) aday üyeler

Grup içi adaylar belirlendikten sonra asil üyeler aday üyelere doğru yönelirler. Bu yönelme sonucunda aday üyelere daha iyi değere sahip asil üye oluşursa, aday üye ile asil üye yer değiştirir. Şekil 12a'da da görüldüğü gibi, asil üyelerin aday üyelere yönelmesinden sonra oluşan yeni değerlere göre aday üyeler değişebilir. Şekil 11b ile Şekil 12a kıyaslandığında, bu yer değiştirme işlemi rahatlıkla fark edilebilir. Şekil 12b'de ise değişimden sonraki her grubun gücü görülmektedir.

(a)	<pre> --grup0-- 28->256 9->267 15->275 22->301 8->309 24->312 16->315 32->316 6->322 1->327 --grup1-- 34->260 27->276 38->285 2->289 20->306 26->318 21->321 29->324 3->343 30->383 </pre>	<pre> --grup2-- 0->235 11->261 33->267 13->293 4->309 36->311 5->314 18->318 37->318 23->349 --grup3-- 7->191 10->193 39->195 31->213 12->219 19->256 17->265 35->272 14->289 25->302 </pre>	(b)
		<pre> power0=290,75 power1=313 power2=295 power3=252,75 </pre>	

Şekil 12: a) Yönelme sonrası oluşan daha uygun değerli üyelerin aday üye seçilmesi b) Her grubun gücü

Bu adımdan sonra artık gruplar arası yarışma başlar. Gelişigüzel bir p_m değeri üretilir ve bu değer belirlediğimiz değerden büyük ise en güçlü iki grup birleştirilir. Yine bu adımda gelişigüzel p_d değeri üretilir. Bu değer de belirlediğimiz değerden küçük olur ise en güçsüz grup silinir. Bu programda p_m %40 olarak p_d ise %1 olarak alınmıştır. Eğer birleşme gerçekleşmezse grup içi yarış adımına tekrar dönlür. Program sonunda tüm gruplar grup1 içinde birleşmiştir ve grup1'in en iyi değerli elemanı problemin çözümüdür. Bir önceki bölümde de bahsedildiği gibi program sonunda asil üyelerin güçlerinin lider olan üyenin gücüne doğru yaklaştığı Şekil 13'te rahatlıkla görülebilir.

```

--grup 1--
39->291  18->347
7->294   34->347
10->295  22->348
35->298  27->349
31->300  16->350
12->302  6->350
17->303  29->350
19->310  24->351
25->317  20->351
14->319  4->353
8->330   36->353
1->332   33->354
28->332  21->356
9->333   37->356
32->337  2->357
38->338  5->358
0->341   26->358
15->342  23->359
11->345  3->362
13->346  30->383

```

Şekil 13: Grupların birleşmesi ve en büyük eleman

SONUÇ

Elektronik ve bilgisayarın gelişmesi ile analitik olarak çözülemeyen birçok problem sayısal olarak çözülmeye başlanmıştır. Bununla birlikte doğal ve sosyal süreç, model ve olaylardan esinlenilerek etkili çeşitli yöntemler geliştirilmiştir.

İnsanların sosyal davranışlarını simüle etme, mühendislikte bazı problemlerin çözümünde yol göstermiştir. Buradan yola çıkarak tabu arama, emperyalist yarışmacı algoritma, parlamenter optimizasyon algoritması ve öğretme-öğrenme tabanlı algoritma gibi yöntemler geliştirilmiştir. Bu yöntemler de diğer sezgisel algoritmalar gibi genel amaçlı ve popülasyon tabanlıdır. Farklı tipteki arama ve optimizasyon problemlerine etkili çözümler sunması beklenmektedir.

Bu makalede ise sosyal tabanlı güncel algoritmalar olan emperyalist yarışmacı algoritma ve parlamenter optimizasyon algoritmalarının nasıl modellendiği anlatılmış ve parlamenter optimizasyon algoritmasının çalışma mantığı örnek bir programda gösterilmiştir. İleriki çalışmalarda matematiksel modelin çıkarılmadığı ya da amaca uygun optimum çözüm yönteminin bulunamadığı problemlerde bu yöntemlerin kullanılması hedeflenmektedir. Hâlihazırda, bu algoritmalar sürekli değerli verilerde sınıflandırma kurallarının keşfi problemlerine uyarlanmış ve farklı veri tabanları için uygulamalara başlanmıştır.

KAYNAKLAR

ABDECHİRİ, M.; K. FAEZ ve H. BAHRAMİ (2010), "Adaptive Imperialist Competitive Algorithm", 9th IEEE International Conference on Cognitive Informatics (ICCI), pp. 940-945.

ABDECHİRİ, M.; K. FAEZ ve H. BAHRAMİ (2010), "Neural Network Learning Based on Chaotic Imperialist Competitive Algorithm", 2nd International Workshop on Intelligent Systems and Applications (ISA), pp.1-5.

ALATAŞ, B. (2007), "Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme", Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü.

ATASHPAZ-GARGARİ, E. ve C. LUCAS (2007), "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition", IEEE Congress on Evolutionary Computation, pp. 4661-4667.

BORJİ, A. (2007), "A New Global Optimization Algorithm Inspired by Parliamentart Political Competitions", Lecture Notes in Computer Science, 4827/2007, pp. 61-71.

BORJİ, A. ve M. Hamidi (2009), "A New Approach to Global Optimization Motivated by Parliamentary Political Competitions", Int. Journal of Innovative Computing Information and Control, vol. 5, no. 6, pp. 1643-1653.

GHALEHPARDAZ, S.L. ve M. Shafiee (2011), "Speed Control of DC Motor Using Imperialist Competitive Algorithm Based on PI-Like FLC", Third International Conference on Computational Intelligence, Modelling and Simulation, pp. 28-33.

KHADEMOLGHORANİ, F. (2011), "An effective algorithm for mining association rules based on imperialist competitive algorithm", Sixth International Conference on Digital Information Management pp. 6-11.

MOVAHED, M.A. ve A.M. Yazdani (2011), "Application of Imperialist Competitive Algorithm in Online PI Controller", Second International Conference on Intelligent Systems, Modelling and Simulation, pp. 83-87.

RAO R.V.; V.J. Savsani ve D.P. Vakharia (2012), "Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems" Information Sciences, vol. 183, no. 1, pp. 1-15,

RAZZAGHPOUR, M ve Rusu, Ana (2011), "Analog circuit optimization via a modified Imperialist Competitive Algorithm", IEEE International Symposium on Circuits and Systems, pp. 2273-2276.

SAYADNAVARD, M.H.; A.T. Haghghat ve M. Abdechiri (2010), "Wireless sensor network localization using Imperialist Competitive Algorithm", 3rd IEEE International Conference on Computer Science and Information Technology, vol. 9, pp.818-822.

TAMİMİ, A.; H. Sadjadian ve H. Omranpour (2010)," Mobile Robot Global Localization using Imperialist Competitive Algorithm", 3rd International Conference on Advanced Computer Theory and Engineering, vol.5, no. V5-524-V5-529, pp. 20-22.

NOTLAR

Bu proje Tunceli Üniversitesi Bilimsel Araştırma Projeleri Birimi (TÜNİBAP) tarafından YLTUB011-14 No'lu proje kapsamında desteklenmektedir.